Parallel Processing with Digital Signal Processing Hardware and Software

Student
Corey V. Swenson

Mentor
Robert L. Jones

Group
Research and Technology

Division
Information and Electromagnetic Technology

Branch
Systems Integration

Parallel Processing with Digital Signal Processing Hardware and Software

Abstract

The assembling and testing of a parallel processing system is described which will allow a user to move a DSP application from the design stage to the execution/analysis stage through the use of several software tools and hardware devices. The system will be used to demonstrate the feasibility of the Algorithm To Architecture Mapping Model (ATAMM) dataflow paradigm for static multiprocessor solutions of DSP applications. The individual components comprising the system are described followed by the installation procedure, research topics, and initial program development.

1.0 Introduction / Background Information

1.1 Multiprocessing of Digital Signal Processing Algorithms

Digital Signal Processing (DSP) systems are used to realize digital filters, compute Fourier transforms, execute data compression algorithms, and many other compute-intensive algorithms. The recent explosion of DSP products on the market reflects the advancements made in DSP technology. The increasing complexity of this DSP technology, especially in real-time systems, has increased computational requirements and created a need for faster, more powerful systems. As a result, government and industry are turning to multiprocessor solutions to meet these needs. In order to take advantage of multiprocessor architectures, the processes of a DSP application must be effectively mapped onto multiple processors. Such mapping procedures are currently in the design stage and are not yet perfected. One mapping procedure, the dataflow paradigm, has been implemented in the Dataflow Design Tool, created by my mentor Robert L. Jones III.

1.2 Dataflow Design Tool

The Dataflow Design Tool for multiprocessor scheduling was developed to facilitate the design of multiprocessor solutions to a number of computational problems, including DSP algorithms and control law. The tool analyzes the computational problem, represented as a dataflow graph, and determines the performance bounds, scheduling constraints, and resource requirements for solving the problem. The tool utilizes the dataflow paradigm to model computational problems. This model uses graphical nodes to represent schedulable computations, direct edges to

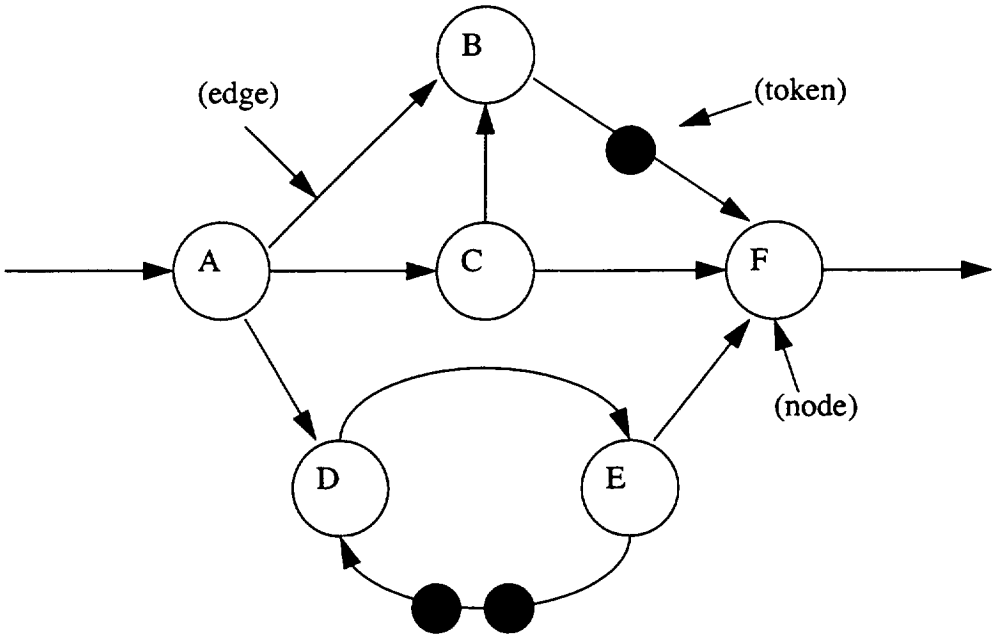

Figure 1. *Dataflow Graph*

describe the dataflow between nodes, and tokens to indicate the presence of data. A dataflow graph is shown in Figure 1.

1.3 Process Scheduling

After an applications algorithm has been modeled, there are two methods for scheduling tasks in a multiprocessor system: static and dynamic. The dynamic scheduling system, shown in the dotted region of Figure 2, has already been implemented and tested with the ATAMM and Dataflow Design Tool on a Generic VHSIC (very high speed integrated circuit) Spaceborne Computer (GVSC). In dynamic scheduling, the tasks to be performed are assigned to a specific processor at *run-time*. Therefore, the system is not dependent on any individual processor, giving it a high degree of fault tolerance. If one processor fails, the algorithm will still execute predictably, only at a degraded level of performance. Dynamic scheduling also gives the system more flexibility. The drawback of dynamic scheduling is it's high overhead. Since it is not known at compile time which processor will be producing a token or which processor will be receiving it, more communication is needed between processors, causing delays and a larger overhead.

The static scheduling system, shown in the solid outline in Figure 2, is the system to be constructed and tested as my LARSS project. In static scheduling, the tasks to be performed are assigned to specific processors at *compile-time*, allowing the programmer to decide which processors perform which tasks. For deterministic DSP algorithms, a priori knowledge can be gathered from the model to make costly decisions about scheduling, communication, and synchronization at compile-time. Thus, making these costly decisions at compile-time minimizes the run-time overhead, allowing more time to be spent doing useful work. However, this system is more rigid and unflexible. It is also less fault-tolerant than dynamic scheduling since the failure of a single processor results in the inability of the system to complete the execution of the algorithm.

The ultimate goal in building and testing these two systems is to thoroughly understand all aspects of each, and through innovative design, create a system which implements a combination

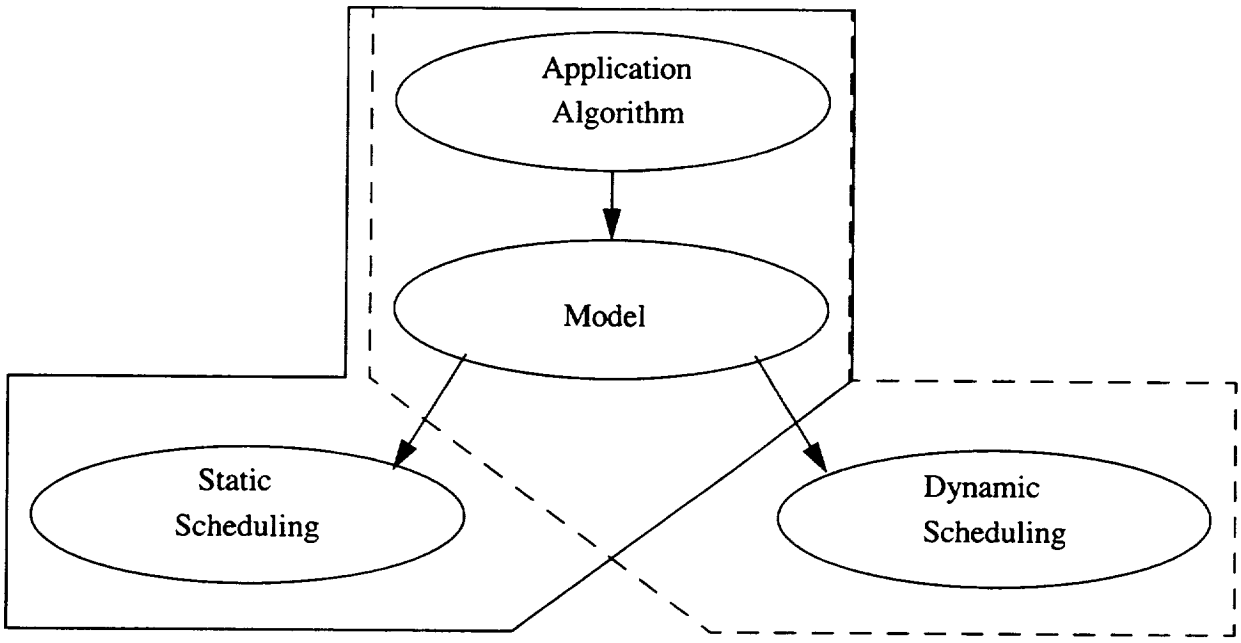

Figure 2. *Process Scheduling*

of static and dynamic scheduling. The resulting system will have a high degree of fault-tolerance and flexibility as well as minimal overhead.

2.0 Summary of Study

2.1 Approach

The multiprocessor system was to utilize in-house models/tools in combination with commercial-over-the-shelf (COTS) software and hardware to realize a suitable testbed. Part of the testbed construction included the selection, installation, setup, and integration of suitable COTS components with state-of-the-art and versatile features that lend themselves to modeling by ATAMM. The in-house models/tools consist of the ATAMM (model) and the Dataflow Design Tool. The COTS tools consist of Hypersignal[1] model capture, automatic code generation, and real-time display; the SPOX[2] operating system's real-time and multiprocessing functions; and Pentek's[3] state-of-the-art digital signal processing boards and debugging software.

When the in-house and COTS components are integrated, the system can be viewed as layers stacked on top of one another to form the system as shown in Figure 3. The application is first created using the Hypersignal graphical software Block Diagram and the code is created with the

1. copyright by Hyperception, Inc. 9550 Skillman, LB 125 Dallas, TX 75243
2. copyright by SPECTRON Microsystems, Inc. 320 Storke Road, Santa Barbara, CA 93117
3. Pentek, Inc. 55 Walnut Street Norwood, NJ 07648

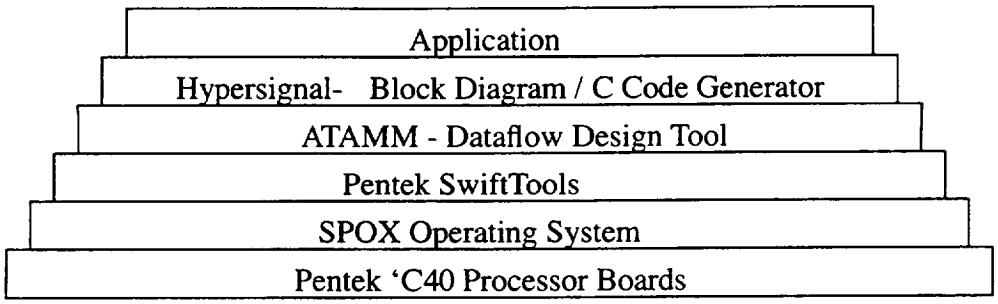

Figure 3. Layered System

C Code Generator. Working with the Block Diagram graphical representation and the dataflow paradigm, the Dataflow Design Tool determines the performance bounds, scheduling constraints, and resource requirements of the application. Using this information, the C code is modified to execute efficiently on the optimal number of processors using Pentek's SwiftTools software. Also, SPOX Operating System (OS) functions may be added to the code to optimize multiprocessor performance. Finally, the code is compiled using the Texas Instruments (TI) optimizing C compiler, downloaded to the Pentek 'C40 boards, and executed.

2.2 Equipment

2.2.1 Hardware Components

Gateway 2000 PC. The Gateway PC contains the software Hypersignal. Through this software the user can create a DSP application, simulate the execution, and see the output. The user can also download the DSP algorithm to a DSP PC/C31 board inside the computer where it is executed. The output may be sent back to the PC and displayed in real-time with the Hypersignal software. The PC can also send an analog signal out or receive an analog signal in via the DSP card. In this way, the PC can receive and display the signal that is processed by a separate computer or computers.

PC/C31 Board. The Loughborough Sound Images' LSI PC/C31 board with A/D D/A daughter module contains a Texas Instruments TMS320C31 processor onto which algorithms can be downloaded via Hypersignal software. The board is also an interface between the Hypersignal software and an external analog signal source. Analog signals come into the PC via the PC/C31 board and are displayed with the RT-2 software.

RadiSys Embedded PC. The RadiSys EPC-5 is a PC/AT compatible embedded CPU module which rests in a card cage along with three Pentek boards. The embedded PC (EPC) has a 66 MHz Intel486 DX2 processor and a VMEbus interface to communicate with the Pentek boards. The EPC contains the SPOX OS and SwiftTools software which allows the user to communicate with the Pentek boards as well as modify and debug C code.

Pentek Model 4202 MIX Baseboard. The 4202 MIX Baseboard is one of the three Pentek boards in the card cage with the RadiSys EPC. The VMEbus, which is part of the card cage, runs along the back of the card cage. The MIXbus runs through the middle of the Pentek boards and is not part of the card cage but part of the Pentek boards. The 4202 converts all of

Pentek's MIX modules into standard VMEbus boards. The 4202 allows communication between the EPC and MIX modules, which are not necessarily connected to the VMEbus.

Pentek Model 4249 Filtered A/D-D/A Converter. The 4249 A/D-D/A converter is the second of the three Pentek boards in the card cage and can support sampling rates of up to 1 MHz. An analog input signal is filtered, sampled by the 12-bit D/A converter, and stored in the 1024 sample FIFO, ready for transfer along the MIXbus to another module. A digital signal received from the MIXbus is put into another 1024 sample FIFO, ready for transfer through the 12-bit D/A converter and low pass filter to the analog output.

Pentek Model 4270 Quad TMS320C40 Digital Signal Processor Board. The 4270 board is the last Pentek board in the card cage. The 4270 contains four Texas Instruments TMS320C40 processors, 4 MBytes of Local SRAM, and 4 MBytes of Global SRAM. Algorithms are downloaded from the EPC onto one or any number of processors on the 4270. The 4270 can send/receive data to/from the 4249 A/D-D/A in it's execution of the algorithm.

Hewlett Packard 3312A Function Generator. The HP Function Generator creates the original signals (sinewaves, squarewaves, etc.) to be used by the system. The Function Generator and other hardware subsystems transmit signals through coaxial cable.

2.2.2 Software Components

Hypersignal for Windows. Hypersignal is a collection of software applications which address many scientific and engineering problems involved in signal processing. This software allows the user of the system to graphically design and simulate an application as well as analyze the results in real-time.

Block Diagram. Block Diagram allows the user to design an application by connecting any number of predefined software building blocks. The user can then compile and run the project, simulating the execution of a real system. Block Diagram also allows the user to build a project using real-time building blocks. These blocks can read in a signal from the outside world, process the signal, then send the signal back out. Many software building blocks are included with the software, but custom blocks may be designed by the user.

Block Wizard. The Block Wizard software simplifies the task of creating new user-defined blocks for use in Block Diagram. The user defines the parameters for the new block and Block Wizard generates the text files necessary for compilation. The user then writes the code to perform the desired task, inserts it into one of the generated files, and compiles the files. The new block function is now ready for use.

C Code Generator. The C Code Generator creates the C source code that represents the algorithm designed with Block Diagram. This code can then be cross-compiled for a particular DSP chip and executed.

RT-2. The RT-2 software provides tools for storing and analyzing real-time signals. The Digital Scopes are used to view an input signal in real time. The displays and controls look identical to an actual oscilloscope. The Spectrum Analyzer displays an input signal in the frequency

domain in real time. The Digital Recorder allows the user to store a real-time signal continuously and then regenerate the signal in real-time at their convenience. Also, the stored waveform can be loaded and viewed by the Graph Analysis software which allows detailed analysis of the signal.

Pentek SwiftTools. SwiftTools is a software development environment for Pentek devices. Through this software, the user can debug and edit C source code. The user can step through the source code, executing one line at a time, while setting breakpoints, viewing register contents, and viewing symbol addresses and values. Through these actions, the user can find design errors within the code. The software also serves as an interface between the EPC and the Pentek boards. This enables the user to download the executable code onto the DSP chips after compilation. The user can stop and start the execution of the program on the DSP chips through the SwiftTools software.

SPOX OS. SPOX is a DSP operating system designed to meet the needs of high-end DSP microprocessors by allowing developers to work with objects relating to signal processing and through general purpose features such as device-independent I/O, interrupt management, and multi-tasking support. SPOX enables the use of the high level C language in real-time signal processing, releasing the full potential of the latest DSP hardware through sophisticated applications.

2.3 System Construction

2.3.1 Installation and Testing

The following Hypersignal software was installed on the Gateway PC:

-Block Diagram
-Block Wizard
-C Code Generator
-RT-2

After installation, Block Diagram examples included with the software were executed on the PC. Several block worksheets were designed and executed using various block functions to ensure the software was loaded properly. Execution of real-time block functions on the 'C31 board required proper configuration, after which real-time block examples were tested.

The following software was installed on the Radisys EPC:

-SwiftTools
-SPOX OS
-TI Optimizing C Compiler
-DSPTools

Prior to running SwiftTools, a configuration program called PNCFG was used to relay information to SwiftTools concerning the Pentek boards used in the system. After configuration, a program was compiled, downloaded to the 'C40 boards, and executed, showing that SwiftTools, the TI Compiler, and the DSPTools were all installed properly. The SwiftTools debugger was used to find errors in a test program for the 4249 A/D-D/A board, which was also shown to be installed correctly. The SPOX software included a sanity check program to test it's installation. Upon execution, the program displayed a message on the screen confirming proper installation.

2.3.2 Installation Delays

Several problems and delays were incurred during installation of the software. For instance, the Unix version of the SPOX manual was received rather than the DOS version, delaying it's installation. Upon receiving the correct SPOX manual, it was realized that the wrong SPOX software version was received as well. A new version was sent but still had the wrong executable files. After receiving new files, the test program finally worked properly. The Hypersignal software has a security button attached to the printer port which was not recognized by the C Code Generator. A new button was sent along with new versions of the C Code Generator and Block Diagram. The new C Code Generator could recognize the new button, but was incompatible with the new Block Generator. A third version of the C Code Generator was sent and worked properly, but the new Block Diagram did not include any real-time block functions to run on the 'C31 board. The SwiftTools PNCFG program was difficult to configure. The Pentek boards had to be removed from the card cage to double check hardware jumper settings. The Pentek representative was contacted several times to find a number of bugs in the setup. Finally, one of the 4270 boards were removed while the other one was configured, allowing the board to be recognized. The other board was inserted but could not be configured, a hardware problem on the board was assumed.

2.3.3 Research

In parallel with the installation and testing of the system components, research of multiprocessing topics was performed to give a better understanding of the system and the purpose of designing it. The topics included the advantages of a multiprocessor system, the classes of architecture, memory configurations, applications, anticipated difficulties, and process scheduling.

2.3.4 4249 and Communication Port Drivers

After installing and successfully testing the 4270 and 4249 boards, drivers were created from the already proven test programs. The test programs contained information regarding registers and initialization needed in any program using the A/D-D/A converter or communication ports. The 4249 driver consists of a routine which takes data from the input FIFO and a routine which sends data to the output FIFO. The com port driver consists of routines which synchronize the sending and receiving of data over com ports. A program was written to implement the two drivers for testing purposes. The first processor receives data from a function generator through the 4249 board, the data is sent sequentially through each processor, the last of which sends the data out through the 4249 board and into the Gateway PC to be displayed by Hypersignal. Although simple, the drivers form the foundations for data transmission and communication between processors.

2.3.5 Final Projects

There are two projects which are expected to be completed in the final week of the program. One project is to create a custom block function using the Hypersignal Block Wizard software which will represent the 4249 driver program. This will enable the user to select the 4249 block

from the function list of Block Diagram, implement it in a block worksheet, and create C code which can be compiled and downloaded to a 'C40 processor without modification in SwiftTools.

The second project consists of moving an application from the design stage to the execution/ analysis stage. After designing the application in Block Diagram and creating the corresponding C code, the application will initially be executed on one processor. By using the Dataflow Design Tool to determine the performance parameters on multiple processors, the C code will be modified to execute on several processors.

3.0 Results

Although no numerical results were obtained from the project, the advantages of constructing the system can be easily seen. After finding all system components to be working properly, example applications can now be created and tested in a very short time. With the 4249 driver program, applications involving the input and output of real signals can be created and executed quickly. The com port driver allows the user to easily synchronize communication between processors, a fundamental aspect of parallel processing. With the completion of the 4249 driver function block in Block Diagram, the user will be able to represent the input and output of the 4249 in graphical form and create C code for applications with the 4249 driver already incorporated.

4.0 Conclusion

The demand for effective mapping procedures for multiprocessor systems was explained with reference to DSP applications and real-time processing. One of these mapping procedures, the dataflow paradigm, has been implemented in the Dataflow Design Tool. The testing of this tool in a dynamic scheduling system was described along with the assembling of a system to test the tool using static scheduling. The individual components comprising this system were listed and their functionality explained. The software/hardware installation was shown to be correct through numerous test programs and the difficulties in integrating the components were recounted. The research of multiprocessing systems performed to gain a better understanding of the system was briefly summarized. Drivers which controlled the communication between processors and signal transmission through the A/D-D/A converter were explained and, finally, the projects still to be completed during the last week of the program were presented.